

AL-TP-1993-0004

AD-A262 289



**AN APPROACH TO ON-LINE ASSESSMENT
AND DIAGNOSIS OF STUDENT
TROUBLESHOOTING KNOWLEDGE**

Nancy J. Cooke
Anna L. Rowe

Rice University
P.O. Box 1892
Houston, TX 77251

DTIC
ELECTE
APR 1 1993
S C D

**HUMAN RESOURCES DIRECTORATE
MANPOWER AND PERSONNEL RESEARCH DIVISION
7909 Lindbergh Drive
Brooks Air Force Base, TX 78235-5352**

Sponsored by:

**Air Force Office of Scientific Research
Bolling Air Force Base, DC 20332**

March 1993

Final Technical Paper for Period June 1992 - August 1992

Approved for public release; distribution is unlimited.

93-06662



98 9 31 101

**AIR FORCE MATERIEL COMMAND
BROOKS AIR FORCE BASE, TEXAS**

ARMSTRONG

LABORATORY

NOTICES

This technical paper is published as received and has not been edited by the technical editing staff of the Armstrong Laboratory.

The views expressed in this paper are those of the authors and do not necessarily reflect official views of the U.S. Air Force or the Department of Defense.

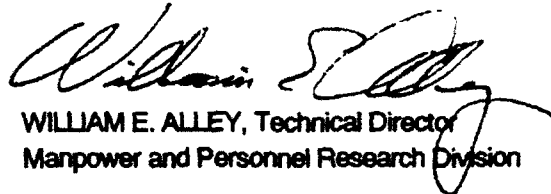
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.



SHERRIE P. GOTT
Senior Scientist
Job Structures Branch



WILLIAM E. ALLEY, Technical Director
Manpower and Personnel Research Division



ROGER W. ALFORD, Lt Col, USAF
Chief, Manpower and Personnel Research Division

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1993	3. REPORT TYPE AND DATES COVERED Final - June 1992 - August 1992	
4. TITLE AND SUBTITLE An Approach to On-Line Assessment and Diagnosis of Student Troubleshooting Knowledge			5. FUNDING NUMBERS C - F49620-90-C-0076 PE - 62205F PR - 7719 TA - 22 WU - 02	
6. AUTHOR(S) Nancy J. Cooke Anna L. Rowe				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Rice University P.O. Box 1892 Houston, TX 77251			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Manpower and Personnel Research Division 7909 Lindbergh Drive Brooks Air Force Base, TX 78235-5352			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AL-TP-1993-0004	
11. SUPPLEMENTARY NOTES Armstrong Laboratory Technical Monitor: Dr. Ellen P. Hall, (210) 536-3570				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Intelligent tutors have the potential to enhance training in avionics troubleshooting by giving students more experience with specific problems. Part of the success of intelligent tutors will be associated with their ability to assess and diagnose the student's knowledge in order to direct pedagogical interventions. The goal of the research program described here is to develop a methodology for assessment and diagnosis of student knowledge of fault diagnosis in complex systems. Along with this broad goal, the methodology should: (1) target system knowledge, (2) provide rich representations of this knowledge useful for diagnosis, (3) be appropriate for real-world complex domains like avionics troubleshooting, and (4) enable assessment and diagnosis to be carried out on-line. In order to meet these requirements a general plan for mapping student actions onto system knowledge is proposed and research from one part of this plan is presented. Results from a Pathfinder analysis on action sequences indicate that action patterns can be meaningfully distinguished for high and low performers and that the patterns reveal specific targets for intervention. Short- and long-term contributions of this work are also discussed.				
14. SUBJECT TERMS Artificial intelligence Diagnostic assessment			15. NUMBER OF PAGES 44	
Intelligent tutoring Student modelling			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

INTRODUCTION.....	1
METHOD.....	9
RESULTS AND DISCUSSIONS.....	10
CONCLUSION.....	10
REFERENCES.....	12
APPENDIX A: The Pathfinder Network Generation Algorithm.....	16
APPENDIX B: Data Flow.....	19

LIST OF FIGURES

FIG NO.		PAGE NO.
1	Steps Involved in Mapping Student Actions Onto System Knowledge.....	6
2	Pretest Network Based on Aggregate Transition Probabilities of the Six Higher Performers.....	13
3	Prettest Network Based on Aggregate Transition Probabilities of the Eight Low Performers.....	14
4	Posttest Network Based on Aggregate Transition Probabilities of the Six High Performers.....	15
5	Posttest Network Based on Aggregate Transition Probabilities of the Eight Low Performers (Link Weights Were Omitted Due To Graph Complexity).....	16

LIST OF TABLES

TABLE NO.		PAGE NO.
1	Intercorrelation Matrix of Four Assessment Measures.....	17
2	Intercorrelation Matrix of Four Measures of Change From Pre to Posttest.....	17
3	Mean Assessment Measures For Experimental and Control Groups on Pre-and Posttests.....	18
4	Number of Nodes and Links For Aggregate Networks of High and Low Performers.....	20

DTIC QUALITY INSPECTED 4

on For	
CRA&I	TAB
ounced	cation
By _____	
Distribution / _____	
Availability Code _____	
Dist	Avail and/or Special
A-1	

PREFACE

This technical paper represents work accomplished under the AF Office of Scientific Research Summer Faculty Research Program. The work was accomplished between June 1992 and August 1992 for the Job Structures Branch of the Manpower and Personnel Research Division, Human Resources Directorate of the Air Force Armstrong Laboratory (AL/HRMJ). The authors wish to thank Drs Sherrie Gott, Ellen Hall, and Bob Pokorny for their assistance and guidance during the course of this project.

This work supports the AL/HRMJ mission to develop methods for assessing complex cognitive skill requirements in Air Force jobs.

AN APPROACH TO ON-LINE ASSESSMENT AND DIAGNOSIS OF STUDENT TROUBLESHOOTING KNOWLEDGE

Nancy J. Cooke

and

Anna L. Rowe

INTRODUCTION

As tasks become more cognitively complex and demand more specialized skill, training issues are increasingly critical. The domain of avionics troubleshooting is a good example of such a task. The cognitive complexity of this task, combined with the personnel downsizing currently faced by the Air Force, make the role of training even more crucial. Personnel will be required to become skilled quickly, and their skill will be required to span a broad range of equipment. In addition, the automatization of many aspects of the troubleshooting task greatly reduces the amount of time spent manually troubleshooting faults. The difficulties associated with the resulting lack of troubleshooting experience are particularly apparent when the automization fails and manual troubleshooting becomes essential. Training programs need to address these rare, yet critical, events.

How can training programs meet these requirements? One approach is through the use of computerized intelligent tutoring systems. These systems enable individuals to spend time learning a skill in a one-on-one environment in which a computer takes on the role of a human tutor. Means and Gott (1988) outline some of the advantages of intelligent tutors, including their ability to provide the student with vast amounts of problem solving experience in a short period of time. Historically, the systems developed for computer-aided instruction were simply on-line displays of a series of written instructions. Individualized instruction, if available, was based on whether or not the student's performance on the instructional material reached a preset criterion. Distinctions were not made on the basis of underlying student knowledge, specific errors that were made, or the possible reasons for those errors (Sleeman & Brown, 1982). One goal of intelligent tutoring systems is to incorporate more individualized instruction based on a detailed assessment of student knowledge and diagnosis of cognitive strengths and weaknesses. Instructional intervention can then be directed at these strengths and weaknesses. The purpose of the work described in this paper is to develop a methodology for the assessment and diagnosis of student knowledge.

The problem of assessment and diagnosis for intelligent tutoring systems has been approached in a number of ways. One approach involves "debugging" a student's knowledge after inferring misconceptions or "mal-rules" from patterns of student errors (e.g., Burton, 1982; Stevens, Collins, & Goldin, 1979). Although this approach has intuitive appeal, there is some evidence that errors are not as systematic as would be implied by underlying misconceptions (Payne & Squibb, 1990). Anderson, Boyle, and Reiser (1985) take a different approach and

model student actions in terms of a set of production rules. These rules are then compared to an ideal student model in order to determine student deficiencies. These approaches and other related ones all attempt to model the student by mapping either errorful actions or all actions onto misconceptions or deficiencies in the student's knowledge. The approaches are also similar in that this mapping is achieved rationally. That is, the ideal model or rules for scoring are constructed through an analysis of domain principles, rather than through an empirical investigation of expert or ideal student behavior. Interestingly, many of the domains studied in intelligent tutoring research have involved rather abstract, academic subjects such as algebra, geometry, and computer programming. These topics tend to lend themselves to a rational analysis because they are well-specified and well-structured problems, typically associated with an organized and well-documented body of knowledge. Although many of the principles and techniques derived from such studies may generalize to other similar domains, it is not clear how such findings can be extended to more complex and concrete domains such as avionics troubleshooting. In many of these domains knowledge acquisition is a prerequisite for tutor development (Psotka, Massey, & Mutter, 1988).

Recently, some assessment work has been carried out in the real-world domain of avionics troubleshooting (Gitomer, 1992; Pokorny & Gott, 1992). Pokorny and Gott (1992) have devised an assessment procedure that identifies general deficits in different types of knowledge (i.e., system, procedural, and strategic) of airmen tasked with troubleshooting technical electronic equipment. In general, they deduct points from these three different knowledge categories depending on the errors that the student makes. Note that this is similar to the debugging approach except that general deficiencies are identified, not specific misconceptions. Gitomer (1992) has developed a related procedure that involves mapping student actions onto these same types of deficits. In both of these cases, a cognitive task analysis that involved knowledge elicitation from subject matter experts was required to determine ideal student behavior.

Likewise, in most real-world domains the first question to be addressed in assessment and diagnosis is exactly what knowledge is necessary to perform the task? Hall, Gott, and Pokorny (1991) have developed a procedure called PARI for analyzing the cognitive requirements of a task for this purpose. The procedure involves a series of structured interviews with subject matter experts (SMEs) in which a specific problem solution is dissected in terms of its precursors, actions, results, and interpretations (i.e., PARI). For instance, a PARI analysis of the avionics troubleshooting domain has indicated that there are several types of knowledge relevant for successful troubleshooting performance. These types include: (1) system (or how it works) knowledge, (2) strategic (or how-to-decide-what-to-do-and-when) knowledge, and (3) procedural (or how-to-do-it) knowledge (Gott, 1989). The assessment and diagnosis of student knowledge of avionics troubleshooting has been guided by the results of this cognitive task analysis in its

focus on these three types of knowledge. Optimally, assessment and diagnosis in this domain would consist of identifying the specific content of student knowledge of these different types and comparing it to SME knowledge. The problem of identifying the content of knowledge is often referred to as knowledge elicitation.

The Problem: Eliciting System Knowledge

Evidence exists that suggests that system knowledge may be the most critical of the three types of knowledge in troubleshooting ill-defined problems in complex systems (Gitomer, 1984). Although there are probably many situations in which an understanding of how the system works is not necessary to perform the task (e.g., automated avionics troubleshooting, making a long distance phone call; Kieras, 1988, Rouse & Morris, 1986), this type of knowledge is assumed to play an important role, at least for problems that are more ill-defined. Though much can be learned about procedural and strategic knowledge from observing the actions of a problem solver, it is much less clear how system knowledge is revealed. Furthermore, the definition of system knowledge or, what many refer to as a mental model, is not completely clear, or at the least, agreed upon (Rouse & Morris, 1986; Wilson & Rutherford, 1989). Despite the lack of a clear definition, research employing the mental model construct is fairly prolific, with different researchers using their own operationalizations of the construct. The different methods for examining mental models can be classified into four categories: 1) accuracy and time measures, 2) interviews, 3) process tracing/protocol analysis, and 4) structural analysis.

Accuracy and time measures. When this method is used, subjects are given a set of problems to solve, and problem solving behavior, measured in terms of time and errors, is examined to make inferences about mental models. Accuracy and/or latency measures of problem solving performance have been used to make inferences about mental models about physics (Clement, 1983; McCloskey, 1983), calculators (Bayman & Mayer, 1984; Halasz & Moran, 1983), electronic circuitry (Gentner & Gentner, 1983), and control panel devices (Kieras & Bovair, 1984). This method is one of the most commonly used in the literature. It is similar to the approach discussed above for debugging student knowledge, both in terms of methodology and limitations. That is, time and errors do not always map neatly onto a specific mental model or misconception (Cooke & Breedin, 1992). As a consequence, most attempts to measure mental models in the literature have combined this basic measure with one or more of the relatively richer measures described below.

Interviews. Interviews to elicit mental models can be more or less structured, with the content and course of the interview being more or less predefined. Unstructured interviews, which do not follow a prespecified format, have been used to capture subjects' mental models of common phenomena in physics (DiSessa, 1983) and of home heat control (Kempton, 1986). Such an interview may also be conducted after problem solving (e.g., McCloskey, 1983). Structured

interviews, on the other hand, follow some sort of prespecified format. The structured interview may focus on: (1) a specific system component--e.g., location, purpose, function (Gitomer, 1984), (2) diagrams--e.g., enumerate concepts, show physical and/or functional relations, designate related components (Gitomer, 1984; Hall, Gott & Pokorny, 1991), or (3) a specific example of system behavior (Stevens, Collins & Goldin, 1979).

Process Tracing/Protocol Analysis. When this method is used, subjects are presented with a problem and are asked to "think aloud" as they solve the problem. Subjects are asked to generally describe their thought processes and to state reasons for their actions. The protocol is subsequently analyzed (i.e., protocol analysis) either to generate hypotheses about mental models or to support or reject a proposed model. Although such verbal reports have been criticized for their reliability and accuracy (e.g., Nisbett & Wilson, 1977), others (Ericsson & Simon, 1984) have attempted to define the conditions under which verbal protocols are appropriate. Nonetheless, this method of measuring mental models is one of the most popular elicitation methods. It has been used to examine mental models of physics (Larkin, 1983; Greeno, 1983; Clement, 1983), calculators (Halasz & Moran, 1983), and heat exchangers (Williams, Hollan, & Stevens, 1983).

Structural Analysis. When this method is used, pairwise proximity estimates for a set of system-relevant items are gathered. These estimates are then submitted to a descriptive multivariate statistical technique (e.g., multidimensional scaling, cluster analysis, or network clustering technique) which reduces the estimates to a simpler form. For example, Kellogg and Breen (1990) used the Pathfinder network structural technique (Schvaneveldt, 1990) to derive and compare user's mental models with an idealized system model. Likewise, Gillan, Breedin, and Cooke (1992) used hierarchical cluster analysis and Pathfinder to compare subjects' mental models of the human-computer interface. In addition, Gitomer (1984) used cluster analysis and multidimensional scaling to compare expert and novice airmen's knowledge organization of an antenna system. One of the strengths of these techniques is that they are able to convey quantitative, as well as qualitative information about mental models.

In summary, four very different types of measurement methods have been used in research on mental models. The different measurement approaches may each provide different sorts of information, making generalizations across studies difficult, if not impossible. In addition, the different approaches have not been evaluated in terms of their respective reliabilities and validities. In general, each of the different methods is likely to have advantages and disadvantages (Cooke, 1992a), and no one method of measuring mental models has received universal acceptance. Therefore the selection of a single optimal method for on-line student assessment is an uncertain enterprise at best. Indeed, the criteria associated with an optimal technique are similarly ill-defined

(Cooke, 1992a). In this paper a pragmatic view is taken in which optimal methods are minimally assumed to elicit knowledge that is relevant to task performance.

Another difficulty associated with using most of these methods for on-line assessment of student system knowledge is that most involve the collection of "extra" data (e.g., verbal reports, similarity ratings) not typically collected in interactions with the tutor. Thus, the use of these methods would entail interruption of the tutoring process to collect data in a task that would most probably seem artificial to the student. The single exception to this limitation is the collection of time and accuracy measures. Here, measures can be automatically collected on-line and used to infer student knowledge. In fact, most of the intelligent tutor approaches to assessment and diagnosis discussed above have relied on this method. Unfortunately, time and accuracy data are impoverished compared to the much richer data obtained from verbal reports and structural analyses. These richer methods go beyond the student's actions, facilitating the jump from actions to the cognitive underpinnings of those actions. Therefore, what is needed is not only a reliable and valid method for measuring system knowledge, but one that can provide rich representations of this knowledge from student actions derived on-line. This is the focus of our project. The goal is to be able to map student actions (both errorful and correct) collected on-line onto a rich representation of student system knowledge. This representation can then be used to assess and diagnose student system knowledge and identify targets for intervention. The domain selected for this project is avionics troubleshooting.

The Plan: Mapping Student Actions onto System Knowledge

Basically, the general problem identified above involves making detailed inferences about a student's system knowledge from that student's actions. One way to dissect this problem is to work backwards from the goal state--system knowledge, to the initial state -- student actions. Interviews, process tracing, and structural analytic methods offer rich representations of system knowledge. However, it is necessary to know which of these methods provides the most reliable and valid measure of system knowledge in the domain of avionics troubleshooting (see Figure 1.1). Therefore, the first subgoal in solving the above problem involves identifying a valid method for eliciting and representing system knowledge required for avionics troubleshooting. Assuming that system knowledge is critical for performance, then a valid method of measuring this knowledge should reveal differences among subjects that correspond to performance differences.

Of course, these techniques require data collected off-line. Therefore, the next subgoal involves determining how to derive this type of data from on-line interactions with the tutor. Can we make use of the data already collected on-line to derive representations of system knowledge? In other words can we identify general relationships between student actions and patterns of system knowledge derived off-line, so that later predictions can be made about system knowledge based on student actions? As previously noted, mapping errors onto a student's understanding can

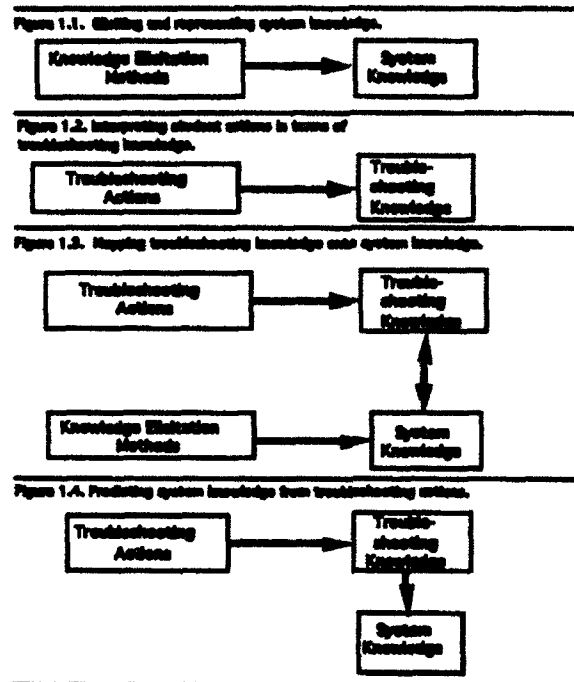


Figure 1. Steps involved in mapping student actions onto system knowledge.

be problematic because actions can be varied and idiosyncratic (e.g., Payne & Squibb, 1990). On the other hand, it is generally assumed that actions are, at least partially, the result of knowledge and that certain patterns of actions reflect specific types of troubleshooting knowledge (Pokorny & Gott, 1992). Gott, Bennett, and Gillet (1986, p 43) label the assumption that "thinking is for the purpose of doing" the theory of technical competence. Perhaps a more stable analysis of student knowledge can be achieved by examining all of the student's actions regardless of whether correct or incorrect. But how do we make sense of all of these actions? What is needed is a means of identifying meaningful patterns or summaries of student actions. A pattern of actions can be thought of as an intermediate representation of student troubleshooting knowledge (see Figure 1.2). Although patterns in student actions are likely to emerge, their meaningfulness is an empirical question. Specifically, do differences revealed in identified action patterns correspond to actual differences in other measures of student performance? Thus, the identification of action patterns and the evaluation of the meaningfulness of these patterns is a second subgoal.

Once meaningful patterns of actions (i.e., troubleshooting knowledge) have been identified, the next subgoal entails mapping these patterns onto patterns of system knowledge (see Figure 1.3). Can we identify patterns of actions that correspond to distinct representations of system knowledge? Of course this step requires the elicitation of both actions and system knowledge from the same subjects. Assuming that the previous subgoals have resulted in meaningful patterns of

actions and representations of system knowledge and assuming that system knowledge underlies actions (at least partially), then some correspondence should emerge. For instance, students who swap a card before checking the data flow to that card may do so for several reasons. This mapping procedure may indicate that students who demonstrate this action pattern tend not to understand the relationship between data flow and signal flow. Finally, if this correspondence does emerge, then it would be possible to make predictions about system knowledge from troubleshooting actions collected on-line, thereby eliminating the extra data collection step (see Figure 1.4). Predictions based on these actions could be evaluated by either implementing them in a tutor and evaluating the tutor or comparing the predictions to those made by SMEs.

The four subgoals represented in Figure 1 comprise the long-term plan associated with the development of a new approach for assessing and diagnosing student system knowledge. The subgoals represented in Figures 1.1 and 1.2 are prerequisites to the later subgoals, but even in isolation, these preliminary steps make important contributions to the general problem of student assessment and diagnosis. More specifically, the first subgoal will identify optimal methods for eliciting system knowledge in the avionics troubleshooting domain. This information is useful for stages of tutor development in which knowledge of this type needs to be elicited from domain experts. In addition, although less efficient than the long-term plan, the best techniques could be used to assess student system knowledge off-line. The second subgoal may also contribute by identifying meaningful action patterns that may be useful in and of themselves in assessing and diagnosing other types of student knowledge (i.e., procedural or strategic knowledge). The remainder of this report focuses on progress made toward the long-term plan, specifically, the subgoal portrayed in Figure 1.2.

Research Progress: Interpreting student actions

The goal of this part of the project is to identify meaningful patterns in students' troubleshooting actions. These patterns are referred to generally as "troubleshooting knowledge," because it is assumed that they are influenced by the three forms of knowledge central to troubleshooting, namely strategic, system, and procedural knowledge. If the resulting action patterns capture troubleshooting knowledge in a meaningful way, then minimally, they should be able to differentiate high and low performers.

One way that action patterns can be derived is through the use of the Pathfinder network scaling procedure. The Pathfinder procedure is a descriptive statistical technique that represents pairwise proximities in a graphical form (Schvaneveldt, 1990; Schvaneveldt, Durso, & Dearholt, 1985; Schvaneveldt, Durso, & Dearholt, 1989). In the graph, concepts or entities are represented as nodes and relations between entities as links between nodes. Each link is associated with a weight that represents the strength of that particular relationship. These weights are based on proximity estimates which can be collected in a number of ways including pairwise relatedness

ratings, co-occurrence of items in a sorting task, or event co-occurrence. Pathfinder networks can have directed links given asymmetrical proximity estimates and unconnected nodes if proximity estimates between an item and all other items exceed a maximum criterion set by the experimenter. It should also be noted that although the links represent semantic relations, the algorithm does not identify the specific relation associated with each link. The Pathfinder procedure determines whether or not to add a link between each pair of nodes. Basically, a link is added if the minimum distance between nodes based on all possible paths (i.e., chains of one or more links) is greater than or equal to the distance indicated by the proximity estimate for that pair. Two parameters, r and q , determine how network distance is calculated and affect the density of the network. Dearholt and Schvaneveldt (1990) provide a detailed discussion of Pathfinder (also see Appendix A).

Pathfinder has several advantages including the fact that it is not constrained to hierarchical configurations like most cluster analysis routines and its ability to represent asymmetrical relations (Dearholt & Schvaneveldt, 1990). In addition, results from several studies indicate that Pathfinder network representations are psychologically meaningful in that they are predictive of recall order and judgment time (Cooke, 1992b; Cooke, Durso, & Schvaneveldt, 1986). Pathfinder networks have, in fact, been used to reliably distinguish skilled and unskilled performers in domains such as air-combat flight maneuvers (Schvaneveldt, Durso, Goldsmith, et al., 1985), computer programming (Cooke & Schvaneveldt, 1988), and interface design (Kellog & Breen, 1990). They have also been used to assess student classroom performance (Goldsmith & Johnson, 1990). In this study the similarity between student and instructor networks was highly correlated ($r = .74$) with final class grade.

The Pathfinder procedure has typically been used to represent knowledge in the form of conceptual or declarative relationships (e.g., Cooke & Schvaneveldt, 1988; Schvaneveldt, Durso, Goldsmith, et al., 1985). However, it has also been used in one case to represent action sequences (McDonald & Schvaneveldt, 1988). In this study McDonald and Schvaneveldt collected co-occurrence frequencies of UNIX commands issued by users who interacted with the system. They used Pathfinder to summarize these data in terms of a network of the most frequently occurring action paths. Thus, because of Pathfinder's ability to represent action sequences and deal with the asymmetrical and nonhierarchical relations typically found in actions, it was selected as a vehicle for interpreting actions in the present study.

Such a representation of actions would be desirable for several reasons beyond the overall goal of mapping actions onto system knowledge. First, on-line assessment in tutors could be achieved by deriving an individual's network from actions taken during problem solving and comparing this network to an "expert" network. The comparison would be based on the number of shared nodes (actions) and links (action sequences) between the two networks. Thus, this

particular comparison results in one value that represents overall level of knowledge. Second, the qualitative nature of the network representation allows a more detailed diagnosis of student troubleshooting knowledge. The Pathfinder network analysis could highlight specific actions and action sequences that are not "expert-like" and that could be targeted for remediation. Likewise, positive aspects of performance (expert-like actions) could be identified and targeted for positive feedback to the student. Thus, one additional benefit of this methodology is that it is capable of providing both quantitative assessment information at a global level and qualitative information at a more detailed level. Finally, because of the bottom-up nature of this approach, the Pathfinder representations may incidentally reveal specific patterns of actions that distinguish high and low performers, but that have not been recognized or verbalized by the SMEs.

METHOD

Actions taken by subjects on a troubleshooting tests described by Nichols, Pokorny, Jones, Gott and Alley (1989) were used to develop Pathfinder networks. In the Nichols et al. study the effects of an intelligent tutoring system called SHERLOCK were examined by comparing the performance of technicians who received both on the job training (OJT) and SHERLOCK training (experimental group) to the performance of technicians who received only OJT (control group).

Subjects.

The subjects were 37 manual avionics shop technicians stationed at one of two AF bases, Langley AFB or Eglin AFB. Supervisors had identified the subjects as being at a beginning or intermediate skill level (3 or 5) and available for the study duration (1 mo.). Five subjects were later dropped from the study: two subjects were transferred, and three subjects were identified as being more skilled than previously determined, leaving a sample of 32 technicians. The subjects were first matched on the basis of a verbal troubleshooting score and a number of other scores (e.g., mechanical and electrical tests). Then members of each matched pair were randomly assigned to either the experimental or control group. The 30 subjects who completed a specific set of three verbal troubleshooting problems were used in the present analyses.

Individual subjects were classified as either high or low performers on each problem based on the score they received from the scoring worksheet (Pokorny & Gott, 1992), the current assessment method in this domain. This score is derived by subtracting a predetermined number of points for each error that the student makes in troubleshooting. For the pretest problem, high performers were defined as those subjects who received a score of 85 or greater, whereas low performers were defined as those subjects who received a score of 35 or less. These cutoffs were arrived at by identification of natural breaks in the frequency distribution of scores. Four of six high performers and three of eight low performers were in the experimental group. Subjects were reclassified as high and low performers based on their performance on the posttest problem. Specifically, subjects were classified as high performers if they received a score of 85 or greater,

and subjects who received a score of 55 or lower were classified as low performers. Interestingly, all of the high performers and only one of the low performers were in the experimental group.

Materials and Procedure.

A brief description of the methodology used by Nichols et al. (1989) follows. All subjects participated in a training period in which they received either OJT or OJT and SHERLOCK. The pre- and posttest measures referred to below were administered before and after this training period, respectively. Four measures were used in the study: 1) the Armed Services Vocational Aptitude Battery, 2) a measure of each subject's previous experience in electronics, 3) pre- and posttest versions of a verbal troubleshooting test, and 4) pre- and posttest versions of a noninteractive troubleshooting test. (Only those subjects stationed at Eglin AFB completed the pretest version of the noninteractive test). In addition, those subjects who received SHERLOCK training completed a tutor report card following the final training session. Only problems from the verbal troubleshooting data were analyzed in the present study.

The verbal troubleshooting test is an individually administered structured problem solving test. The test begins with the examiner describing a fault that has occurred. The subject then attempts to isolate the fault and repair the equipment through a series of recursive action-result steps. In each step the subject specifies an action he/she would take and the reason for taking that particular action. The examiner responds by informing the subject of the action's effect on the equipment, and requests the subject's inference concerning equipment operation based on that effect. The cycle continues until the problem is solved, the one hour time limit expires, or the subject gives up. Thus, although subjects are not working on actual equipment, they have to make use of all of the technical data that they would require if they were troubleshooting real equipment.

Six pretest and four posttest verbal troubleshooting problems were administered by Nichols et al. Only the data from three problems were used in the present analyses, specifically pretest 1, pretest 2, and posttest 1. The complete analysis described below was conducted on data from pretest 2 and posttest 1 because these problems were comparable in terms of type and difficulty. The pretest 1 problem was primarily analyzed to determine the optimal coding scheme.

RESULTS AND DISCUSSION

A coding scheme for students' actions was developed using the data from the pretest 1 problem (see Appendix B). This scheme was then applied to and modified slightly for the remaining two problems, referred to herein as pretest and posttest. The purpose of the scheme was to be able to classify discrete actions into meaningful action units that could be represented as nodes in a Pathfinder network. The main categories of actions for both problems included equipment checks, data flow tests, signal flow tests, and swaps. The most abstract level of categorization was used unless the same action would, in some cases, result in a pass and in others, a fail. In this case, the lower, more specific level of abstraction was used. Using this

decision rule, for each problem an action unit was associated with one and only one troubleshooting outcome. The resulting coding schemes consisted of 63 action units/nodes categories for the pretest and 62 action units/nodes for the posttest problem.

Transition probabilities for all pairs of actions (in both directions) were calculated for individual subjects by dividing the frequency with which specific action transitions (e.g., swap UUT followed by check DMM fuse) occurred by the frequency with which the first item in the sequence occurred. For example, if swap UUT occurred twice and was followed by check DMM fuse on one of those occasions then the transition probability would be 0.5. Note that these are first-order transitions only. Higher order transitions (i.e., the probability of swap UUT followed by check DMM fuse either immediately or with one or more actions intervening) were considered, but not used because the immediate transitions were considered to be the most meaningful. Transition probabilities were also calculated across groups of subjects using frequencies summed across all subjects in the group. For instance, transition probabilities were calculated for the high and low performers for each of the two troubleshooting problems.

The four matrices of transition probabilities (high and low performers, pre- and posttest) were submitted to the Pathfinder network network scaling technique (Schvaneveldt, 1990). Figures 2 and 3 illustrate the pretest problem network representations resulting from the high and low performers' probabilities, respectively. Figures 4 and 5 illustrate the posttest problem network representations resulting from the high and low performers' probabilities respectively. Details of these networks will be discussed below in the section on diagnosis.

Assessment

One of the major questions to be asked of this approach is whether Pathfinder networks of actions can distinguish high and low performers for the purposes of assessment. In this study the subjects' score for each problem derived using the scoring worksheet is assumed to be the "true score" indication of their performance on that problem. Therefore, to answer the above question one can look at the correlation between an assessment measure derived from Pathfinder networks and the score derived from the scoring worksheet procedure. To assess students using Pathfinder, for each problem an ideal or expert network can be compared to the network representation of each nonexpert individual. The C measure (Goldsmith & Davenport, 1990) provides a quantitative index of network similarity that can be used for this purpose. This measure is based on proportion of shared nodes and links in two networks. It ranges from 0 (low similarity) to 1 (high similarity). For the pre- and posttest problems, the networks based on the aggregate actions of the six highest performers were used as ideals for that problem. The remaining nonexperts were evaluated in terms of these standards. Note that the use of the six highest performers as the ideal greatly restricts the range of data for the remaining nonexperts on which the correlations were based. This procedure was necessary because there were only incomplete data available for SMEs, the obvious

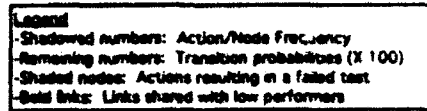
choice for the ideal. Thus, it should be kept in mind that the correlations reported here may be underestimated due to this constraint.

The correlations between troubleshooting scores and this network similarity measure for the 24 nonexperts in each problem are presented in Table 1. In addition two other assessment measures that were related to the network similarity measure were calculated and included in the analysis to aid in distinguishing relevant from irrelevant aspects of the Pathfinder-based measure. One of these measures was derived from a correlation of action frequencies (i.e., the frequency with which each action unit occurred) associated with an individual's protocol and action frequencies associated with the aggregate high-performer protocol. Thus, this measure should be high to the extent that the nonexpert performed the same actions as the high-performers the same number of times. It should overlap with the Pathfinder network similarity measure in that they both take shared actions into account. However, the Pathfinder measure includes information on action sequences, whereas the action frequency measure includes frequency of individual actions. Finally, the second other measure was the total number of actions that each subject executed (i.e., number of steps to solution).

Examination of Table 1 indicates that the Pathfinder similarity measure is predictive of troubleshooting scores for the pretest ($r(22) = .57, p < .01$), but not for the posttest ($r(22) = .26$). However, the action frequency measure is predictive of the score for both the pre- ($r(22) = .65, p < .01$) and the posttest ($r(22) = .76, p < .01$). Other significant correlations indicate that the two measures of Pathfinder similarity and action frequency are highly intercorrelated, as was predicted. However, at least for the pretest, both measures seem to independently account for a portion of the variance. The correlation between the troubleshooting score and the action frequency measure remains significant when the Pathfinder similarity measure is partialled out ($r(21) = .53, p < .01$). Also, the correlation between the troubleshooting score and the Pathfinder similarity measure is marginally significant when the action frequency measure is partialled out ($r(21) = .39, p < .07$).

Another way of looking at these data is to compute change scores for subjects from pretest to posttest and correlate these scores. Because only 20 of the 24 nonexperts were classified as nonexperts for both tests, data were analyzed for only these 20 subjects. The intercorrelation matrix for these change scores is presented in Table 2. As might be expected from the previous analysis, the troubleshooting score change was highly correlated with both the change in Pathfinder similarity ($r(18) = .51, p < .05$) and the change in action frequency ($r(18) = .55, p < .05$).

Taken together, these results suggest that the types of actions subjects perform and the frequency with which they perform them are predictive of both the pre- and posttest scores. In addition, the specific sequence in which actions are executed is predictive of the pretest scores. As will be discussed below, there was a much wider range of actions performed by the low



13

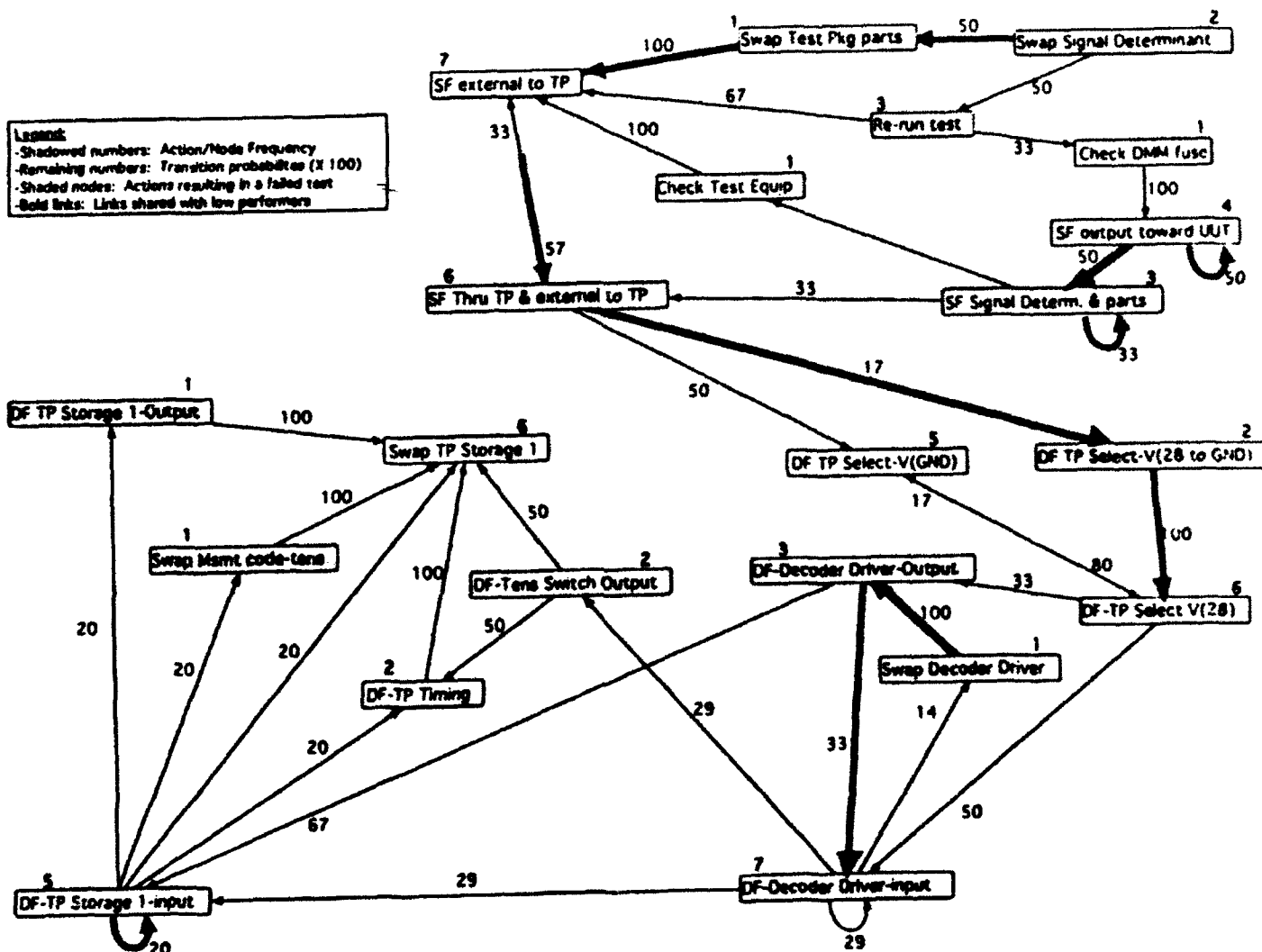


Figure 4. Posttest network based on aggregate transition probabilities of the six high performers.

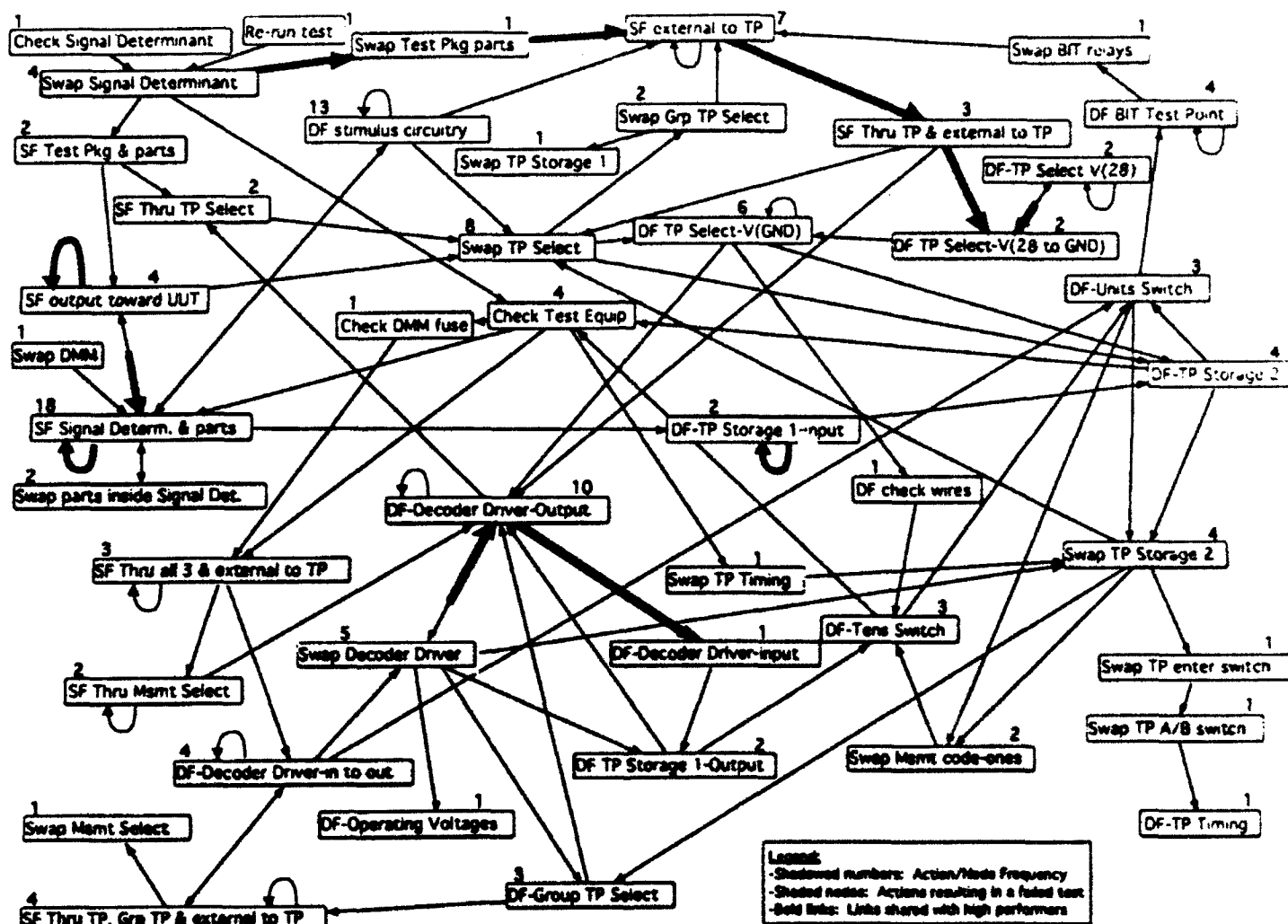


Figure 5. Posttest network based on aggregate transition probabilities of the eight low performers (link weights are omitted due to graph complexity).

Table 1. Intercorrelation matrix of four assessment measures. (VT score = verbal troubleshooting score; PF sim = similarity of Pathfinder network with expert network; ActFreq=correlation of action frequencies with expert action frequencies; No.Act = number of actions)

Table 1a.

Pretest Measures				
	1	2	3	4
1. VT score	1.00	.57**	.65**	.38
2. PF sim		1.00	.47*	.38
3 ActFreq			1.00	.30
4. No.Act-				1.00

Table 1b.

Posttest Measures				
	1	2	3	4
1. VT score	1.00	.26	.76**	-.35
2. PF sim		1.00	.55**	.22
3 ActFreq			1.00	-.17
4. No.Act				1.00

* $p < .05$; ** $p < .01$

Table 2. Intercorrelation matrix of four measures of change from pre- to posttest. (VT score = verbal troubleshooting score; PF sim = similarity of Pathfinder network with expert network; ActFreq=correlation of action frequencies with expert action frequencies; No.Act = number of actions)

Intercorrelations of Change From Pre- to Posttest

	1	2	3	4
1. VT score change	1.00	.51*	.55*	-.075
2. PFsim change		1.00	.49*	.37
3. ActFreq change			1.00	.11
4. No.Act change				1.00

* $p < .05$; ** $p < .01$

performers in the posttest compared to the pretest which may have overwhelmed any predictive power of sequential variation.

Finally, the assessment measures can also be compared in terms of their ability to discriminate subjects in the experimental and control groups. The mean scores of experimental and control subjects for the pretest and posttest are presented in Table 3. As should be expected, there were no pretest differences between experimental and control groups. Interestingly, the only significant difference between these two groups at posttest is for the Pathfinder similarity measure ($t(22) = 2.07, p < .05$). Subjects in the experimental condition had networks that were more similar to the ideal network than did subjects in the control condition. The lack of a significant verbal troubleshooting score difference between the two groups is most likely due to the restriction of range that occurred by eliminating the six highest performers on the posttest. The fact that Pathfinder accounts for experimental vs. control differences, but not the action frequency measure, suggests that subjects who were trained on SHERLOCK learned more expert-like action sequences than those who were not.

Table 3. Mean assessment measures for experimental and control groups on pre- and posttests. (VT score = verbal troubleshooting score; PF sim = similarity of Pathfinder network with expert network; ActFreq=correlation of action frequencies with expert action frequencies; No.Act = number of actions)

Measure	Pretest Mean	Posttest Mean
<u>VTscore</u>		
Experimental	42.00	68.00
Control	47.00	59.00
<u>PFsim</u>		
Experimental	.05	.07
Control	.05	.04
<u>ActFreq</u>		
Experimental	.38	.41
Control	.33	.24
<u>No.Act</u>		
Experimental	12.60	15.80
Control	11.50	16.60

In sum, this procedure seems to identify meaningful action patterns. Assessment in this domain (i.e., avionics troubleshooting) is currently carried out using the scoring worksheet (Pokorny & Gott, 1992) and, as demonstrated above, an assessment measure based on Pathfinder action patterns corresponded to that of the scoring worksheet. Although this particular subgoal does not entail diagnosis of student knowledge, one of the purported benefits of the Pathfinder

analysis is its ability to offer information beyond the mere assessment of student knowledge. In this section diagnostic implications of the Pathfinder analyses are discussed. The main question here is does Pathfinder highlight specific strengths and weaknesses in students' knowledge that can be targeted for intervention? The analysis that follows entails identifying the strengths and weaknesses of the low performers as a whole, although an identical analysis could be performed at an individual level.

Diagnosis and Intervention

The Pathfinder networks for the high and low performers differed both quantitatively and qualitatively. Some of the quantitative differences between individuals and high-performers were captured in the network similarity measures described above. General quantitative differences between the two groups can be seen in terms of the number of nodes and links present in the networks of the high and low performers. The high performers' networks had fewer nodes (i.e., actions) than the low performers' networks, especially at posttest (see Figures 2 through 5 and Table 4). In other words, the high performers as a group executed fewer distinct actions than the low performers, indicating a less varied repertoire of actions across all high performers for this problem. High performers seem to agree on the relevant actions for this problem in comparison to low performers. Although the low performers at posttest executed over twice as many distinct actions as the high performers, they shared all but one of the high performer's actions. Thus, at posttest the low performer's applied a wide repertoire of actions as a group, including actions that are expert-like. These results suggest that the low performers as a group have knowledge about a wide variety of actions by posttest, yet they do not seem to understand when these actions apply. Interestingly, the subjects in the experimental group executed fewer distinct actions (35) than those in the control group (48). Thus, SHERLOCK may be effective in teaching students the conditions under which various actions apply.

What do these differences indicate in terms of diagnosis and intervention? First, the six pretest nodes in the high performers' network that were not contained in the low performers' network consisted of signal flow and data flow tests. In addition, at pretest, low performers executed 11 actions (corresponding to 11 extra nodes) high performers did not, seven of which were data flow and signal flow tests. At posttest, half of the additional actions executed by low performers were data flow and signal flow tests and half were swaps. Thus, these errors of omission and commission indicate that intervention in these particular cases should be targeted at learning the appropriate data flow and signal flow actions. A more detailed target may be derived by a focus on individual nodes.

In addition to having fewer nodes than low performers, high performers' networks also had fewer links than the low performers' at both pre- and posttest (see Table 4). This is to be expected given the fact that fewer nodes necessarily implies fewer links. However, the number of

links per node does not differ greatly for the four networks. For each node there are approximately 2 links per node (range = 1.8 to 2.1) across the four networks. However, the number of links shared between the high and low performers increased slightly from pre- to posttest, suggesting that the low-performers began demonstrating action sequences more like those of the high performers. This pattern is verified by the C measure of similarity between the networks of the low and high performers at pre- ($C = .04$) and posttest ($C = .07$). Although the resulting C values were relatively low, they do indicate that the low performers became more like the high performers at posttest. For instance, even the low performers at posttest had learned to conduct the signal flow and data flow tests which the high performers used at posttest to pinpoint the fault. Thus, the low performers learned more expert-like sequences of actions, given training.

Table 4. Number of nodes and links for aggregate networks of high and low performers.

	Number of Nodes		Number of Links	
	Pretest	Posttest	Pretest	Posttest
High performers	23	21	44	38
Low performers	28	45	52	94
Shared	17	20	7	11

The networks of the high and low-performers also differed in some more global ways. First, the high performers (both tests) appeared to follow a rule about the general sequence of actions which were taken: 1) general checks outside of the test package, including visual checks, equipment checks, and swaps, 2) signal flow tests inside the test station, 3) data flow tests of components inside the test station and 4) swapping. Low performers, on the other hand, did not closely follow this rule and instead committed violations in this general sequence. For example, some low performers moved from data flow tests inside the test package to general checks outside of the test package. This trend was observed both for pre- and posttest networks.

Second, the low performers exhibited what may be termed a meaningless action sequence at both pre- and posttest, whereas high performers did not. For example, after completing a signal flow or data flow check which indicated that the component was functional, some low performers chose to swap the component anyway. The high performers did not exhibit meaningless action sequences such as these.

CONCLUSIONS

The results obtained from the work completed thus far are promising in that they indicate that meaningful patterns of actions can be identified using the Pathfinder network scaling procedure. This result achieves the subgoal indicated in Figure 1.2. The network patterns are also meaningful

in the sense that they can differentiate high and low performers as defined by the scoring worksheet. In addition, the Pathfinder networks reveal qualitative differences between high and low performers that are suggestive of targets for intervention (e.g., data flow and signal flow tests). Finally, this bottom-up approach to knowledge elicitation resulted in general action patterns that may not have been verbalized in a typical knowledge elicitation interview (i.e., the general sequence of high-performers: checks outside, signal flow tests inside, data flow tests, swaps). These results are even more promising when the source of the ideal or expert network used to make these comparisons is considered. Specifically, high-performers were used here as the ideal. An even better ideal would probably result from the use of recognized SMEs. In addition, the use of subjects with more expertise would widen the range of performance, which would likely result in enhanced assessment and diagnostic capabilities.

The next step of this project is the evaluation of different measures of system knowledge (the subgoal represented in Figure 1.1). The longer-term goals include the mapping of system knowledge onto action patterns and prediction of system knowledge based on this mapping.

The short term (one year) contributions of this work include:

1. A method of generating network representations of student actions and an evaluation of this method.
2. An alternative to, or extension of, existing methods for assessing student troubleshooting knowledge on-line.
3. A method for targeting specific concepts or strategies associated with overall knowledge strengths or deficits.
4. A method or set of methods that have been determined to be optimal for eliciting and representing the system knowledge of students.

The longer-term contributions of this work are:

1. A procedure for on-line assessment and diagnosis of student's system knowledge which involves mapping action patterns onto deficits or proficiencies in system knowledge.
2. A procedure which summarizes actions (errorful and correct) in terms of a rich representation of student knowledge that lends itself to qualitative analysis useful for diagnosis and intervention.
3. An assessment and diagnosis procedure that targets the complex domain of avionics troubleshooting.
4. A methodology that can be applied to the problem of eliciting knowledge from SMEs for tutor development.
5. A general test of the assumption that system knowledge underlies troubleshooting actions.

REFERENCES

- Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. Science, 228, 456-462.
- Bayman, P. & Mayer, R.E. (1984). Instructional manipulation of users mental models for electronic calculators. International Journal of Man-Machine Studies, 20, 189-199.
- Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman and J. S. Brown (Eds.), Intelligent Tutoring Systems (pp. 157-183). London: Academic Press.
- Clement, J. (1983). A conceptual model discussed by Galileo and used intuitively by physics students. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 325-340). Hillsdale, NJ: Erlbaum.
- Cooke, N.J. (1992a). A Taxonomy and Evaluation of Knowledge Elicitation Techniques. Lockheed Engineering and Sciences Corporation Technical Report. Houston, TX: Human-Computer Interaction Laboratory
- Cooke, N. J. (1992b). Predicting judgment time from measures of psychological proximity. Journal of Experimental Psychology: Learning, Memory, and Cognition, 18, 640-653.
- Cooke, N. J. & Breedin, S. D. (1992 manuscript under review). Constructing naive theories of motion on-the-fly. Journal of Experimental Psychology: Learning, Memory, and Cognition.
- Cooke, N. M., Durso, F. T., & Schvaneveldt, R. W. (1986). Recall and measures of memory organization. Journal of Experimental Psychology: Learning, Memory and Cognition, 12, 538-49.
- Cooke, N. M., & Schvaneveldt, R. W. (1988). Effects of computer programming experience on network representations of abstract programming concepts. International Journal of Man-Machine Studies, 29, 407-427.
- Dearholt, D. W., & Schvaneveldt, R. W. (1990). Properties of Pathfinder networks. In R. Schvaneveldt (Ed.), Pathfinder associative networks: Studies in knowledge organization (pp. 267-277). Norwood, NJ: Ablex.
- DiSessa, A.A. (1983). Phenomenology and the evolution of intuition. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 15-33). Hillsdale, NJ: Erlbaum .
- Ericsson, K.A., & Simon, H.A. (1984). Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.
- Gentner, D., & Gentner, D.R. (1983). Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 99-129). Hillsdale, NJ: Erlbaum .
- Gillan, D.J., Breedin, S.D., & Cooke, N.M. (1992). Network and multidimensional representations of the declarative knowledge of human-computer interface design experts. International Journal of Man-Machine Studies, 36, 587-615.
- Gitomer, D.H. (1984). A cognitive analysis of a complex troubleshooting task. Unpublished doctoral dissertation, University of Pittsburgh, Pittsburgh, PA.

- Gitomer, D.H. (1992). Foundations of HYDRIVE. Presentation given at the Basic Job Skills Progress Review, Brooks AFB, San Antonio, April 7-9.
- Goldsmith, T. E., & Davenport, D. M. (1990). Assessing structural similarity of graphs. In R. Schvaneveldt (Ed.), Pathfinder Associative Networks: Studies in Knowledge Organization (pp. 75-87). Norwood, NJ: Ablex.
- Goldsmith, T. E., & Johnson, P. J. (1990). A structural assessment of classroom learning. In R. Schvaneveldt (Ed.), Pathfinder Associative Networks: Studies in Knowledge Organization (pp. 241-253). Norwood, NJ: Ablex.
- Gott, S. P. (1989). Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies. In E. Z. Rothkopf (Ed.), Review of Research in Education, Vol 15, (pp. 97-169). Washington D.C.: American Educational Research Association.
- Gott, S. P., Bennett, W., & Gillet, A. (1986). Models of technical competence for intelligent tutoring systems. Journal of Computer-Based Instruction, 13, 43-46.
- Greeno, J.G. (1983). Conceptual entities. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 227-252). Hillsdale, NJ: Erlbaum.
- Halasz, F.G., & Moran, T.P. (1983). Mental models and problem solving in using a calculator. In A. Janda (Ed.), Human factors in computing systems: Proceedings of CHI 1983 conference (pp. 212-216). New York: Association for Computing Machinery.
- Hall, E.M., Gott, S.P., & Pokorny, R.A. (1991). A Procedural Guide to Cognitive Task Analysis: The PARI Methodology. Air Force Technical Report, in preparation. Brooks AFB, TX: Air Force Armstrong Laboratory, Human Resources Directorate, Manpower and Personnel Division.
- Kellogg, W.A., & Breen, T.J. (1990). Using Pathfinder to evaluate user and system models. In R. W. Schvaneveldt, (Ed.), Pathfinder Associative Networks: Studies in Knowledge Organization. Norwood, NJ: Ablex.
- Kempton, W. (1986). Two theories of home heat control. Cognitive Science, 10, 75-90.
- Kieras, D.E. (1988). What mental model should be taught: Choosing instructional content for complex engineered systems. In J. Psotka, L.D. Massey, & S.A. Mutter (Eds.), Intelligent Tutoring Systems: Lessons Learned (pp. 85-118). Hillsdale, NJ: Erlbaum.
- Kieras, D.E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. Cognitive Science, 8, 255-273.
- Larkin, J.H. (1983). The role of problem representation in physics. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 75-98). Hillsdale, NJ: Erlbaum.
- McCloskey, M. (1983) Naive theories of motion. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 299-323). Hillsdale, NJ: Erlbaum.
- McDonald, J. E., & Schvaneveldt, R. W. (1988). The application of user knowledge to interface design. In R. Guindon (Ed.), Cognitive Science and its Applications for Human-Computer Interaction (pp. 289-338). Hillsdale, NJ: Erlbaum.

- Means, B., & Gott, S. P. (1988). Cognitive task analysis as a basis for tutor development: Articulating abstract knowledge representations. In J. Psotka, L. D. Massey, and S. A. Mutter (Eds.), Intelligent Tutoring Systems: Lessons Learned (pp. 35-57). Hillsdale, NJ: Erlbaum.
- Nichols, P.D., Pokorny, R., Jones, G., Gott, S.P., & Alley, W.E. (1989). Effective Instructional Processes Within an Intelligent Tutoring System. Unpublished manuscript. Brooks AFB, TX: Air Force Armstrong Laboratory, Human Resources Directorate, Manpower and Personnel Division
- Nisbett, R.E., & Wilson, T.D. (1977). Telling more than we can know: Verbal reports on mental processes. Psychological Review, 84, 231-259.
- Norman, D.A. (1983). Some observations on mental models. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 7-14). Hillsdale, NJ: Erlbaum.
- Payne, S. J., & Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of error. Cognitive Science, 14, 445-481.
- Pokorny, R., & Gott, S. (1992, in preparation). The Evaluation of a Real-World Instructional System: Using Technical Experts as Raters. Air Force Technical Report. Brooks AFB, TX: Air Force Armstrong Laboratory, Human Resources Directorate, Manpower and Personnel Division.
- Psotka, J., Massey, L. D., & Mutter, S. A. (1988). Knowledge acquisition. In J. Psotka, L. D. Massey, and S. A. Mutter (Eds.), Intelligent Tutoring Systems: Lessons Learned (pp. 15-19). Hillsdale, NJ: Erlbaum.
- Rouse, W.B., & Morris, N.M. (1986). On looking into the black box: Prospects and limits in the search for mental models. Psychological Bulletin, 100, 349-363.
- Schvaneveldt, R. (Ed.), (1990). Pathfinder Associative Networks: Studies in Knowledge Organization. Norwood, NJ: Ablex.
- Schvaneveldt, R. W., Durso, F. T., & Dearholt, D. W. (1985). Pathfinder: Scaling with Network Structures. Memorandum in Computer and Cognitive Science, MCCS-85-9, Computing Research Laboratory, New Mexico State University.
- Schvaneveldt, R. W., Durso, F. T., & Dearholt, D. W. (1989). Network structures in proximity data. In G. H. Bower (Ed.), The Psychology of Learning and Motivation: Advances in Research and Theory (Vol. 24, pp. 249-284). New York: Academic Press.
- Schvaneveldt, R. W., Durso, F. T., Goldsmith, T. E., Breen, T. J., Cooke, N. M., Tucker, R. G., & DeMaio, J. C. (1985). Measuring the structure of expertise. International Journal of Man-Machine Studies, 23, 699-728.
- Sleeman, D. & Brown, J. S. (1982). Introduction: Intelligent Tutoring Systems. In D. Sleeman and J. S. Brown (Eds.), Intelligent Tutoring Systems (pp. 1-11). London: Academic Press.
- Stevens, A., Collins, A., & Goldin, S.E. (1979). Misconceptions in student's understanding. International Journal of Man-Machine Studies, 11, 145-156.

- Williams, M.D., Hollan, J.D., & Stevens, A.L. (1983). Human reasoning about a simple physical system. In D. Gentner & A.L. Stevens (Eds.), Mental Models (pp. 131-153). Hillsdale, NJ: Erlbaum.
- Wilson, J.R., & Rutherford, A. (1989). Mental models: Theory and application in human factors. Human Factors, 31, 617-634.
- Young, R.M. (1981). The machine inside the machine: User's models of pocket calculators. International Journal of Man-Machine Studies, 15, 51-85.

Appendix A

The Pathfinder Network Generation Algorithm

The Pathfinder procedure takes pairwise proximity estimates for a set of items and generates a graph structure in which the items are represented as nodes and relations between items as links between nodes. Links connecting nodes are determined on the basis of the pattern of proximity estimates. Each link is associated with a weight that represents the strength of that particular link. Weights are the original proximity estimates associated with item pairs. With symmetric distance matrices Pathfinder will produce networks with undirected links. However, Pathfinder networks can have directed links given asymmetrical estimates and can be unconnected if proximity estimates between an item and all other items exceed a maximum criterion set by the experimenter. The major diagonal in the data matrix represents the distance between an object and itself. This distance is usually 0, but Pathfinder will handle non-zero entries on the diagonal by creating links from the node to itself (loops) in the network. Data derived from transition probabilities may lead to such non-zero entries for the diagonal

The data for Pathfinder may be in the form of similarities, dissimilarities, probabilities, or distances. The data may be collected from records of events (e.g., actions taken in problem solving), eye movements, or more typically, from concept similarity ratings or co-occurrence in concept sorting. For example, suppose three subjects (Tom, Michelle, and Doug) were asked to rate all pairs of the following four entities in terms of relatedness (1=highly related, 6=not related):

- (1) DMM
- (2) Group Test Point Select Card
- (3) UUT
- (4) Test Point Select Card

The hypothetical data can be formatted in a symmetrical matrix as follows. Rows and columns correspond to the four entities:

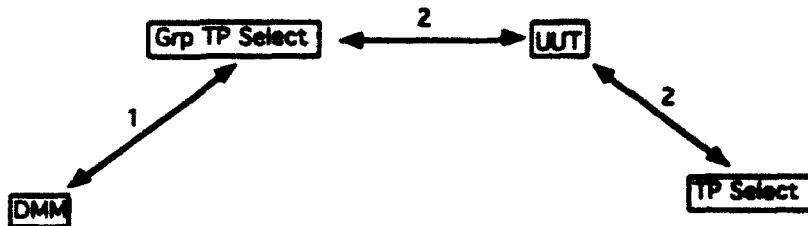
Tom:	Michelle:	Doug:
0 6 2 4	0 6 4 4	0 5 4 3
6 0 5 3	6 0 1 1	5 0 2 3
2 5 0 5	4 1 0 3	4 2 0 1
4 3 5 0	4 1 3 0	3 3 1 0

The Pathfinder procedure determines whether or not to add a link between each pair of nodes. A link is added if the minimum distance between nodes based on all possible paths (i.e., chains of one or more links) is greater than or equal to the distance indicated by the proximity estimate for that pair.

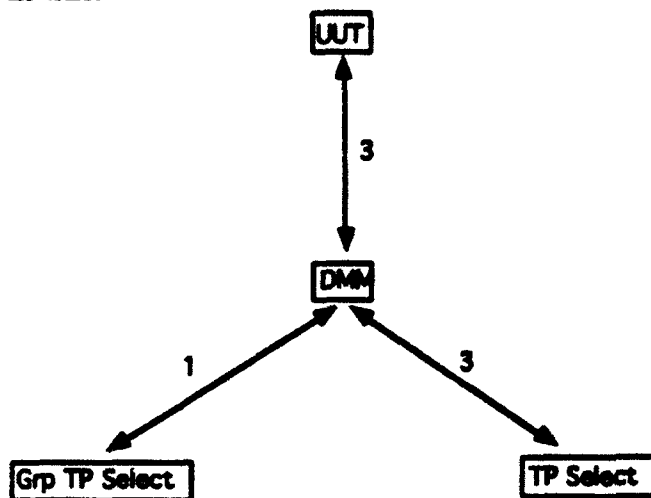
Pathfinder uses two parameters, q and r , to determine the calculation of this network distance. The q -parameter constrains the number of links traversed in paths in the network. The r -parameter defines the metric used for computing the path length in terms of the Minkowski metric, so $r = 1$ corresponds to the city block metric and $r = 2$ corresponds to the Euclidean metric. When $r = \text{infinity}$, path length equals the maximum weight (i.e., distance estimate) of the links that make up the path, and thus only ordinal assumptions need to be made about the data. Varying these two parameters results in networks of differing complexity; however, it is always the case that the links of simpler networks are completely contained within more complex networks. The simplest network results from setting r to infinity and q to the number of items (or nodes) minus one.

The Pathfinder networks ($r = \infty$, $q = 3$) based on the relatedness ratings given by the three hypothetical data sets are shown below. Note that the network structures of Michelle and Doug are highly similar; the two networks are structurally alike except for a link between Grp TP Select and TP Select seen in Doug's network but not in Michelle's. On the other hand, Tom's network is different from both Michelle's and Doug's. Tom did not see DMM as central, whereas both Michelle and Doug did. In addition, Tom's network is "chain-like", whereas Michelle and Doug's networks are not.

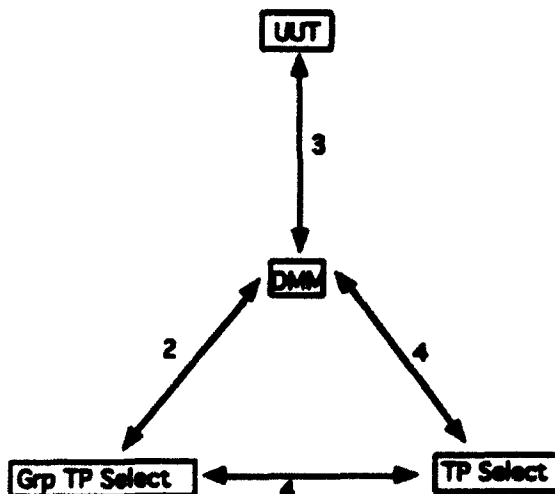
Tom:



Michelle:



Doug:



In addition to a qualitative comparison of the networks, a quantitative comparison can be made using the C statistic (Goldsmith & Davenport, 1990). This is a measure of shared links for matching nodes across two different networks. C indicates the strength of relationship between two networks and ranges in value from 0 (not related) to 1 (related). The first step in calculating C is determining the proportion of shared links for a particular node across two networks. This is accomplished by calculating the ratio of the intersection of links emanating from that node to the union of links from that node across the two networks. This proportion is calculated for all nodes across the two networks. C is the averaging ratio of shared links across the nodes in the two networks. The calculations of C between A, B, and C's networks are illustrated below.

Tom and Michelle

Step 1: $1/3 + 1/2 + 0/3 + 0/2 = .833$

Step 2: $.83 / 4 = .21$

Tom and Doug

Step 1: $1/3 + 1/3 + 0/3 + 0/3 = .667$

Step 2: $.667 / 4 = .17$

Doug and Michelle:

Step 1: $3/3 + 1/2 + 1/1 + 1/2 = 3$

Step 2: $3/4 = .75$

Thus, Doug and Michelle's networks are fairly strongly related, with $C = .75$. Tom's network, on the other hand, is not as strongly related to either Michelle's or Doug's with $C = .21$ and $.17$, respectively.

Appendix B

- *: Action units used for the pretest problem
- #: Action units used for the posttest problem

DATA FLOW

1.0 DF check of Test Point select card *

- 1.1 V(28V) #
 - 1.11 oscscope
 - 1.12 voltage
 - 1.13 ohm
- 1.2 V(GND) #
 - 1.21 oscscope
 - 1.22 voltage
 - 1.221 Off the active path
 - 1.23 ohm
- 1.3 V(28 V to GND) #
 - 1.31 oscscope
 - 1.32 voltage
 - 1.321 Off the active path
 - 1.33 ohm

2.0 DF check of Group Test Point Select card

- 2.1 V(28V) *
 - 2.11 oscscope
 - 2.12 voltage
 - 2.13 ohm
- 2.2 V(GND) *
 - 2.21 oscscope
 - 2.22 voltage
 - 2.23 ohm
- 2.3 V(28 V to GND) *
 - 2.31 oscscope
 - 2.32 voltage
 - 2.33 ohm

3.0 DF Check of Measurement Select Card *

- 3.1 A DMM GND
 - 3.11 oscscope
 - 3.12 voltage
 - 3.13 ohm
- 3.2 B DMM GND
 - 3.21 oscscope
 - 3.22 voltage
 - 3.23 ohm
- 3.3 V (28V)
 - 3.31 oscscope
 - 3.32 voltage
 - 3.33 ohm

4.0 DF check of Decoder Driver

4.1 Input *

- 4.11 oscscope
- 4.12 voltage
- 4.13 ohm

4.2 Output

- 4.21 V(28V) *
 - 4.211 oscscope
 - 4.212 voltage
 - 4.2121 Off the active path
 - 4.213 ohm
- 4.22 V(GND) *
 - 4.221 oscscope
 - 4.222 voltage
 - 4.2221 Off the active path
 - 4.223 ohm
- 4.23 V(28 V to GND) *
 - 4.231 oscscope
 - 4.232 voltage
 - 4.233 ohm

4.4 By ohm check (28V output to GND output) *

4.5 From input to output *

- 4.51 oscscope
- 4.52 voltage
 - 4.521 Off the active path
- 4.53 ohms
 - 4.531 Off the active Path

5.0 DF check of TP Storage 2 *

5.1 Input

- 5.11 measurement code
 - 5.111 oscscope
 - 5.112 voltage
 - 5.113 ohm
 - 5.1131 Off the active path
- 5.12 from TP timing (Enter, Reset, A Enter, B Enter)
 - 5.121 oscscope
 - 5.122 voltage
 - 5.123 ohm

5.2 Output

- 5.21 oscscope
- 5.22 voltage
- 5.23 ohm

5.3 From input to output

6.0 DF check of TP Storage 1 *

6.1 Input

- 6.11 measurement code
 - 6.111 oscscope
 - 6.112 voltage
 - 6.1121 Off the active path
 - 6.113 ohm
 - 6.1131 Off the active path

6.12 from TP timing (Enter, Reset, A Enter, B Enter)
6.121 oscscope
6.122 voltage
6.123 ohm

6.2 Output #
6.21 oscscope
6.22 voltage
6.23 ohm

7.0 DF check of TP Timing * #
7.1 Input
7.11 A/B
7.111 oscscope
7.112 voltage
7.113 ohm
7.12 Enter
7.121 oscscope
7.122 voltage
7.123 ohm

7.2 Output
7.21 oscscope
7.22 voltage
7.23 ohm

7.3 From input to output

8.0 DF check Units Switch * #
8.1 Input
8.11 oscscope
8.12 voltage
8.13 ohm
8.2 Output
8.21 oscscope
8.22 voltage
8.23 ohm

9.0 DF check Tens Switch * #
9.1 Input
9.11 oscscope
9.12 voltage
9.13 ohm
9.2 Output
9.21 oscscope
9.22 voltage
9.23 ohm

10.0 DF check Enter Switch * #
10.1 oscscope
10.2 voltage
10.3 ohm

11.0 DF check A/B Switch * #
11.1 oscscope
11.2 voltage
11.3 ohm

12.0 DF check Measurement Select Switch Output * #

12.1 A DMM

- 12.11 oscscope**
- 12.12 voltage**
- 12.13 ohm**

13.0 DF check Operating Voltages * #

13.1 Off the active path

45.0 DF check wires * #

47.0 DF check BIT Test Point * #

48.0 DF check stimulus circuitry * #

SIGNAL FLOW

14.0 SF Wires * #

14.1 DMM to A1

- 14.12 short**
- 14.13 ohm**
- 14.13 voltage**

14.2 A1 to A12

- 14.21 short**
- 14.22 ohm**
- 14.23 voltage**

14.3 A12 to A13

- 14.31 short**
- 14.32 ohm**
- 14.33 voltage**

14.4 A13 to TP

- 14.41 short**
- 14.42 ohm**
- 14.43 voltage**

14.5 Off the active path

- 14.51 short**
- 14.52 ohm**
- 14.53 voltage**

15.0 SF thru Test Point Select Card * #

15.1 short

15.2 ohm

15.21 Off the active path

15.3 voltage

16.0 SF thru Group Test Point Select Card * #

16.1 short

16.2 ohm

16.3 voltage

17.0 SF thru Measurement Select Card * #

17.1 short

17.2 ohm

17.3 voltage

18.0 SF thru Test Point Select & Group Test Point Select * #

- 18.1 short
- 18.2 ohm
- 18.3 voltage

19.0 SF thru Measurement Select & Group Test Point Select * #

- 19.1 short
- 19.2 ohm
- 19.3 voltage

20.0 SF thru all three cards * #

- 20.1 short
- 20.2 ohm
- 20.3 voltage

21.0 SF thru all three cards & Test Package * #

- 21.1 short
- 21.2 ohm
- 21.3 voltage

22.0 SF thru all three cards, & external to Test Point Select * #

- 22.1 short
- 22.2 ohm
- 22.3 voltage

23.0 SF thru Test Point Select, Group Test Point Select, & external to Test Point Select * #

- 23.1 short
- 23.2 ohm
- 23.3 voltage

24.0 SF thru Test Point Select & external to Test Point Select * #

- 24.1 short
- 24.2 ohm
- 24.3 voltage

25.0 SF external to Test Point Select * #

- 25.1 short
- 25.2 ohm
- 25.3 voltage

26.0 SF Test Package/Test Package Parts * #

- 26.1 short
- 26.2 ohm
- 26.21 Off the active path
- 26.3 voltage

27.0 SF Signal Determinant (UUT & MSA) & parts inside * #

- 27.1 short
- 27.2 ohm
- 27.3 voltage

28.0 SF DMM fuse * #

28.1 short

28.2 ohm

28.3 voltage

46.0 SF output towards UUT * #

CHECKS

29.0 Check Signal Determinant (UUT & MSA) * #

29.1 visual

29.2 part number (P/N)

30.0 Check Test Equipment * #

30.1 visual

30.3 setup/settings

31.0 Check DMM * #

31.1 visual

31.2 part number (P/N)

31.3 setup

32.0 Check DMM fuse * #

32.1 Simpson ohmmeter

32.2 oscscope

32.3 swap

32.4 visual check

SWAP

33.0 Swap Signal Determinant (UUT & MSA) * #

34.0 Swap parts inside Signal Determinant * #

35.0 Swap DMM * #

36.0 Swap DF Component

36.1 Decoder Driver * #

36.2 Test Point Storage 2 * #

36.3 Test Point Storage 1 * #

36.4 Test Point Timing * #

36.5 Measurement Code-Ones * #

36.6 Measurement Code-Tens * #

36.7 BIT Relays Test Point * #

36.8 TP enter switch * #

36.9 TP A/B switch * #

37.0 Swap Test Point Select Card * #

37.1 Off the active path

38.0 Swap Group Test Point Select Card * #

39.0 Swap Measurement. Select Card * #

40.0 Swap Test Package Parts * #

41.0 Swap Other * #

RESEAT

42.0 Reseat Card * #

42.1 Reseat Measurement Select

42.2 Reseat Group Test Point Select

42.3 Reseat Test Point Select

42.4 Reseat Decoder Driver/DF Component

OA/FI

43.0 OA/FI on RAG Drawer * #

RERUN TEST

44.0 Re-run Test/Re-enter instructions * #